(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 909 074 A1

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
**14.04.1999 Bulletin 1999/15**

(51) Int Cl.[6]: **H04L 29/06**

(21) Application number: **98306998.0**

(22) Date of filing: **01.09.1998**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **12.09.1997 US 927382**

(71) Applicant: **LUCENT TECHNOLOGIES INC.**
**Murray Hill, New Jersey 07974-0636 (US)**

(72) Inventors:
- **Coss, Michael John**
  **Bridgewater, New Jersey 08807 (US)**
- **Sharp, Ronald L.**
  **Califon, New Jersey 07830 (US)**
- **Majette, David L.**
  **Bernardsville, New Jersey 07924 (US)**

(74) Representative:
**Watts, Christopher Malcolm Kelway, Dr. et al**
**Lucent Technologies (UK) Ltd,**
**5 Mornington Road**
**Woodford Green Essex IG8 OTU (GB)**

## (54) Methods and apparatus for a computer network firewall with multiple domain support

(57) The invention provides improved computer network firewalls which include one or more features for increased processing efficiency. A firewall in accordance with the invention can support multiple security policies, multiple users or both, by applying any one of several distinct sets of access rules. The firewall can also be configured to utilize "stateful" packet filtering which involves caching rule processing results for one or more packets, and then utilizing the cached results to bypass rule processing for subsequent similar packets. To facilitate passage to a user, by a firewall, of a separate later transmission which is properly in response to an original transmission, a dependency mask can be set based on session data items such as source host address, destination host address, and type of service. The mask can be used to query a cache of active sessions being processed by the firewall, such that a rule can be selected based on the number of sessions that satisfy the query. Dynamic rules may be used in addition to pre-loaded access rules in order to simplify rule processing. To unburden the firewall of application proxies, the firewall can be enabled to redirect a network session to a separate server for processing.

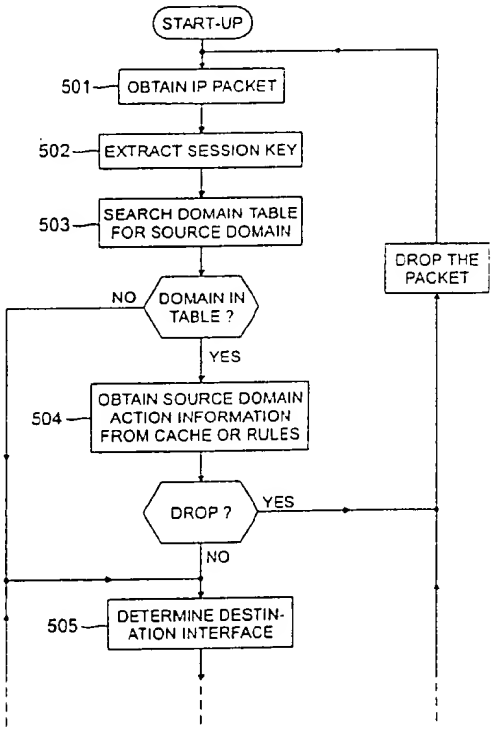


FIG. 5A

EP 0 909 074 A1

Printed by Jouve, 75001 PARIS (FR)

(Cont. next page)

## Description

### Field of the Invention

[0001] This invention relates to the prevention of unauthorized access in computer networks and, more particularly, to firewall protection within computer networks.

### Background of the Invention

[0002] In computer networks, information is conventionally transmitted in the form of packets. Information present at one site may be accessed by or transmitted to another site at the command of the former or the latter. Thus, e.g., if information is proprietary, there is a need for safeguards against unauthorized access. To this end, techniques known as packet filtering, effected at a network processor component known as a firewall, have been developed and commercialized. At the firewall, packets are inspected and filtered, i.e., passed on or dropped depending on whether they conform to a set of predefined access rules. Conventionally, these rule sets are represented in tabular form.

[0003] Typically, a firewall administrator allows broad access which is consented to from one side of the firewall to the other, but blocks transmissions in the opposite direction which are not part of an active network session. For example, "inside" company employees may have unrestricted access through the firewall to an "outside" network such as the Internet, but access from the Internet is blocked unless it has been specifically authorized. In addition to such a firewall at a corporate boundary to the Internet, firewalls can be interposed between network domains, and can also be used within a domain to protect sub-domains. In each case, different security policies may be involved.

[0004] In certain complex network protocols, separate, additional network sessions are required from the outside back to the user. One such complex protocol is employed by a service known by the trade name "RealAudio." Without special measures, the request for the separate session will be blocked by the firewall.

[0005] For such complex protocols, separate "proxy" processes have been developed to run concurrently on the firewall processor on behalf of the user. Proxy processes have also been developed for other special-purpose applications, e.g., to perform services such as authentication, mail handling, and virus scanning.

[0006] In the interest of maximizing the number of sessions which can run concurrently, since the capacity of a firewall processor to support concurrent processes is limited, it is desirable to minimize the need for proxy processes on the firewall. Such minimization is desirable further in the interest of over-all transmission rate, as passage of incoming data through separate processes tends to slow transmission down.

### Summary of the Invention

[0007] The present invention provides techniques for implementing computer network firewalls so as to improve processing efficiency, improve security, increase access rule flexibility, and enhance the ability of a firewall to deal with complex protocols. In accordance with a first aspect of the invention, a computer network firewall is able to support (a) multiple security policies, (b) multiple users, or (c) multiple security policies as well as multiple users, by applying any one of several distinct sets of access rules for a given packet. The particular rule set that is applied for any packet can be determined based on information such as the incoming and outgoing network interfaces as well as the network source and destination addresses.

[0008] In accordance with a second aspect of the invention, a computer network firewall can be configured to utilize "stateful" packet filtering which improves performance by storing the results of rule processing applied to one or more packets. Stateful packet filtering may be implemented by caching rule processing results for one or more packets, and then utilizing the cached results to bypass rule processing for subsequent similar packets. For example, the results of applying a rule set to a particular packet of a network session may be cached, such that when a subsequent packet from the same network session arrives in the firewall, the cached results from the previous packet are used for the subsequent packet. This avoids the need to apply the rule set to each incoming packet.

[0009] In accordance with a third aspect of the invention, a computer network firewall authorizes or prevents certain network sessions using a dependency mask which can be set based on session data items such as source host address, destination host address, and type of service. The dependency mask can be used to query a cache of active sessions being processed by the firewall, to thereby identify the number of sessions that satisfy the query. The query may be associated with an access rule, such that the selection of that particular rule is dependent on the number of successful matches to the query.

[0010] In accordance with a fourth aspect of the invention, a computer network firewall may make use of dynamic rules which are added to a set of access rules for processing packets. The dynamic rules allow a given rule set to be modified based on events happening in the network without requiring that the entire rule set be reloaded. Exemplary dynamic rules include a "one-time" rule which is only used for a single session, a time-limited rule which is used only

[0028] The security policies can be represented by sets of access rules which are represented in tabular form and which are loaded into the firewall by a firewall administrator. As illustrated in Fig. 3, such a table can provide for categories including rule number, designations of source and destination hosts, a designation of a special service which can be called for in a packet, and a specification of an action to be taken on a packet. Special services can include proxy services, network address translation, and encryption, for example. In Fig. 3, the categories "Source Host," "Destination Host" and "Service" impose conditions which must be satisfied by data included in a packet for the specified action to be taken on that packet. Other conditions can be included, and such conditions need not relate to data included in the packet. For example, application of a rule can be made conditional on the time of day or day of the week.

[0029] When a category provided for in the rule table is irrelevant in a certain rule, the corresponding table entry can be marked as a "wild card." This can apply to any one or any combination of the categories. In Fig. 3 and elsewhere, an asterisk (*) is used for wild card entries. "FTP" stands for "file transfer protocol."

[0030] In rule processing for a packet, the rules are applied sequentially until a rule is found which is satisfied by the packet (or until the rule table is exhausted, in which case the packet is dropped). For a packet to satisfy a rule, each condition included in the rule must be met. For example, with reference to Fig. 3, a packet from source host A to destination host D and representing mail will be dropped under Rule 20. The following is a more detailed list of exemplary rule set categories in accordance with the invention. The first five category names correspond to the categories shown in Fig. 3.

| Category Name | Description |
|---|---|
| Rule Number | Number of rule within domain. Rule numbers do not have to be unique but should generally represent a single service, such as FTP |
| Source Host | Source host group identifier or IP address |
| Destination Host | Destination host group identifier or IP address |
| Service | Service group or protocol/destination port/source port |
| Action | Rule action, e.g., "pass," "drop" or "proxy" |
| Notify on Drop | If "yes," an Internet Control Message Protocol (ICMP) message is sent out if action is "drop" |
| Cache Timeout | Number of seconds of inactivity before session entry is removed from cache |
| Reset Session | If "yes," for TCP sessions, send TCP reset to both ends of connection upon cache timeout |
| Rule Timeout | Number of seconds of inactivity before rule is removed from rule list |
| Start Period | Start active period for rule |
| End Period | End active period for rule |
| Kill Session at End of Period | If "yes" then any sessions authorized by this rule will be killed at the end of the time period |
| Dependency Mask | Dependency mask name |
| In Interface | Interface name to match on reception |
| Out Interface | Interface name to which packet is sent |
| Audit Session | Audit record generation. If "yes" then audit record is generated at the beginning and again at the end of the session. |
| Alarm Code | Alarm code value used to tie rule to particular alarms |
| Source Host Map Group | IP address or host group containing map-to host IP addresses |
| Source Host Map Type | Type of mapping to be performed, e.g., "pool" or "direct" |
| Destination Host Map Group | IP address or host group containing map-to host IP addresses |
| Destination Host Map Type | Type of mapping to be performed, e.g., "pool" or "direct" |
| Service Map Group | Service group containing map-to destination port numbers or the destination port. Protocol and source port in a referenced service group are ignored. |
| Service Map Type | Type of mapping to be performed, e.g., "pool" or "direct" |
| Max Use Total Count | Maximum number of times this rule may be used. The rule is removed after the limit is reached. |
| Max Use Concurrent Count | Maximum number of sessions authorized by this rule which may be active at a given time. The rule is inactive until the count falls below the designated value. |
| Copy to Address | Address of application to which a copy of packet is sent. Used for session captures. |

the cache, the rule set for the source domain is searched for a match; if a match is found in the rules and if the action is not "drop," the process continues with step 505; if a match is found in the rules and the action is "drop," a corresponding entry is included in the cache, the packet is dropped, and the process returns to step 501; if no match is found in the rules, the packet is dropped and the process returns to step 501;

505: the destination interface is determined using the local area network (LAN) address of the packet, and, if the source domain rule specifies a destination interface, using that destination interface and a routing table;

506: using the destination interface and the destination address of the packet, the destination domain is determined; if the destination domain is not found, or if the destination domain matches the domain just checked, the process skips to step 508;

507: cache look-up and, if required, rule set look-up for the destination domain are carried out in a manner analogous to that employed for the source domain in step 504;

508: if a rule that applies to the packet calls for an address change, e.g., to a proxy or for insertion of one packet into another ("tunnel option"), the process returns to step 505 for processing based on the changed destination;

509: if the packet was not processed with respect to any domain, the packet can be dropped, as a firewall owner has no interest in supporting communications between interfaces which are not subject to any access rules;

510: with all actions having resulted in "pass," the packet is sent out the appropriate network interface.

**[0035]** For convenient linking of each network interface to a domain, a domain table is used. In cases where an interface is shared by multiple domains, an address range is included. This is illustrated by Fig. 6 which shows non-overlapping address ranges.

**[0036]** Fig. 7 illustrates domain table processing as performed in steps 503 and 506 described above, including the following steps:

701: the domain table is searched for a match of the interface name;

702: if a matching table entry is found, and if the IP address range is present in the matching table entry, the packet address is checked as to whether it is within the range; if so, the specified domain is selected; otherwise, the search continues with the next table entry;

703: if the end of the table is reached without a match having been found, no action is taken.

3 Dependency Mask

**[0037]** For protocols of the type which require a separate, additional network session from the outside back to the user, such as, for example, the protocol employed by RealAudio, a rule can include a condition or mask that allows a connection back to a user, but only if there is a proper forward connection concurrently active, i.e., a connection in which the source and destination addresses are interchanged. As a result, there is no need for a separate or proxy application on the firewall.

**[0038]** A dependency mask in accordance with the invention can define a query directed to the session cache. A match is determined by matching all fields defined in the mask with the corresponding fields in the cache. Empty fields within the mask are not used for comparison.

**[0039]** A dependency mask may be defined in a rule for the first packet of a network session, using (a) information in the packet, (b) the source interface for that packet and (c) one or several dependency conditions that must be met for the packet to pass. When such a first packet has been processed by the firewall, a corresponding entry is made in the cache.

**[0040]** Fig. 8 shows rules with a dependency mask ("hit count") in a format similar to that of Fig. 3. Special symbols are included for certain host designations, namely (i) a "dot" symbol (.) calling for inclusion of packet data of the corresponding category, and (ii) a caret symbol (^) calling for inclusion of packet data from a certain different category instead. "Hit count" indicates the number of matches which must be found in the cache for the specified action to be taken. For example, in the dependency mask named "realaudio," a count of 1 is used for passing UDP packets provided one requisite TCP session is active. In the dependency mask "telnet," a count of 10 is used to drop packets to prevent overloading of resources.

**[0041]** Fig. 9 illustrates dependency mask processing including the following steps:

901: the packet is obtained and the session key is extracted;

902: the process steps through the rule set entries; if no match is found with a given rule, the process advances to the next rule in the set; if no match is found by the time the rule set is exhausted, the packet is dropped; if a match is found and the dependency mask field is null, the process skips to step 905;

903: the packet and interface information may be included in the formation of a cache search structure, e.g., a query; if a user authentication flag is set in the dependency mask, the corresponding flag is set in the cache search

1002: action associated with the packet is determined by looking in the appropriate session cache or, if not found in the cache, in the appropriate rule set; if the action is "pass" or "proxy," packet processing continues; if the action is "drop," the packet is dropped;
1003: if the action indicates a proxy application supported locally on the firewall, the packet is sent up the protocol stack to an awaiting proxy application;
1004: if the action indicates a remote proxy, the packet's destination address is replaced with the address of the remote proxy; if configured, the destination port can be changed as well; the original packet header data is recorded in the session cache along with any changed values;
1005: the packet is routed to the remote proxy server.

[0050] Fig. 10B illustrates processing at the remote proxy, subsequent to step 1005, including the following steps:

1006: the packet is received in the remote proxy server application;
1007: the remote proxy contacts the firewall for the original session key for the packet;
1008: the remote proxy application uses the original session key to perform its function, such as dropping the connection based on its own security model, performing the requested service, or contacting the original destination address on behalf of the user; if the remote proxy is using single reflection, the process skips to step 1011;
1009: the remote proxy application contacts the firewall over the encrypted channel to request dual reflection capability;
1010: the firewall determines a new destination port number that will guarantee uniqueness of the connection from the server; the firewall passes this new port number and the original session key back to the proxy application;
1011: the remote proxy application requests permission from the firewall for a connection from itself to the original destination;
1012: the firewall loads a dynamic rule to perform this action;
1013: the remote proxy sends the packet to the firewall; based on the dynamic rule loaded in step 1012, the firewall forwards the packet to the original destination; in the case of dual reflection, the proxy uses the destination port which was determined by the firewall in step 1010, and, as the packet passes through the firewall, the IP header values are changed back to the original values.

[0051] All future packets associated with the same session are processed alike, except that steps 1007 and 1009-1012 can be skipped. This is because the same dynamic rules apply for the life of the session.

[0052] The invention can be implemented in a wide variety of applications. For example, the invention may be used to provide improved firewall performance in a dial-up access gateway. Another exemplary embodiment of the invention may be implemented in a distributed manner with a first portion of the firewall resident in the network and a second portion of the firewall resident in a set-top box, computer or other user terminal in a home or business. The latter embodiment can allow the firewall techniques of the invention to provide, for example, parental control of Internet and video access in the home. These and the other above-described embodiments of the invention are intended to be illustrative only. Numerous alternative embodiments within the scope of the following claims will be apparent to those skilled in the art.

## Claims

1. A method for validating a packet in a computer network, comprising the steps of:

   deriving a session key for said packet;
   selecting at least one of a plurality of security policies as a function of the session key; and
   using the selected at least one of the security policies in validating said packet.

2. The method of claim 1 wherein the session key includes items derived from header information appended to data in said packet.

3. The method of claim 1 wherein the session key includes at least one item from the set consisting of (i) a source address, (ii) a destination address, (iii) a next-level protocol, (iv) a source port associated with a protocol, and (v) a destination port associated with the protocol.

4. The method of claim 1 wherein the session key includes at least one item from the set consisting of (i) an Internet protocol (IP) source address, (ii) an IP destination address, (iii) a next-level protocol, (iv) the source port associated



XOCID: <EP__0909074A1_I_>

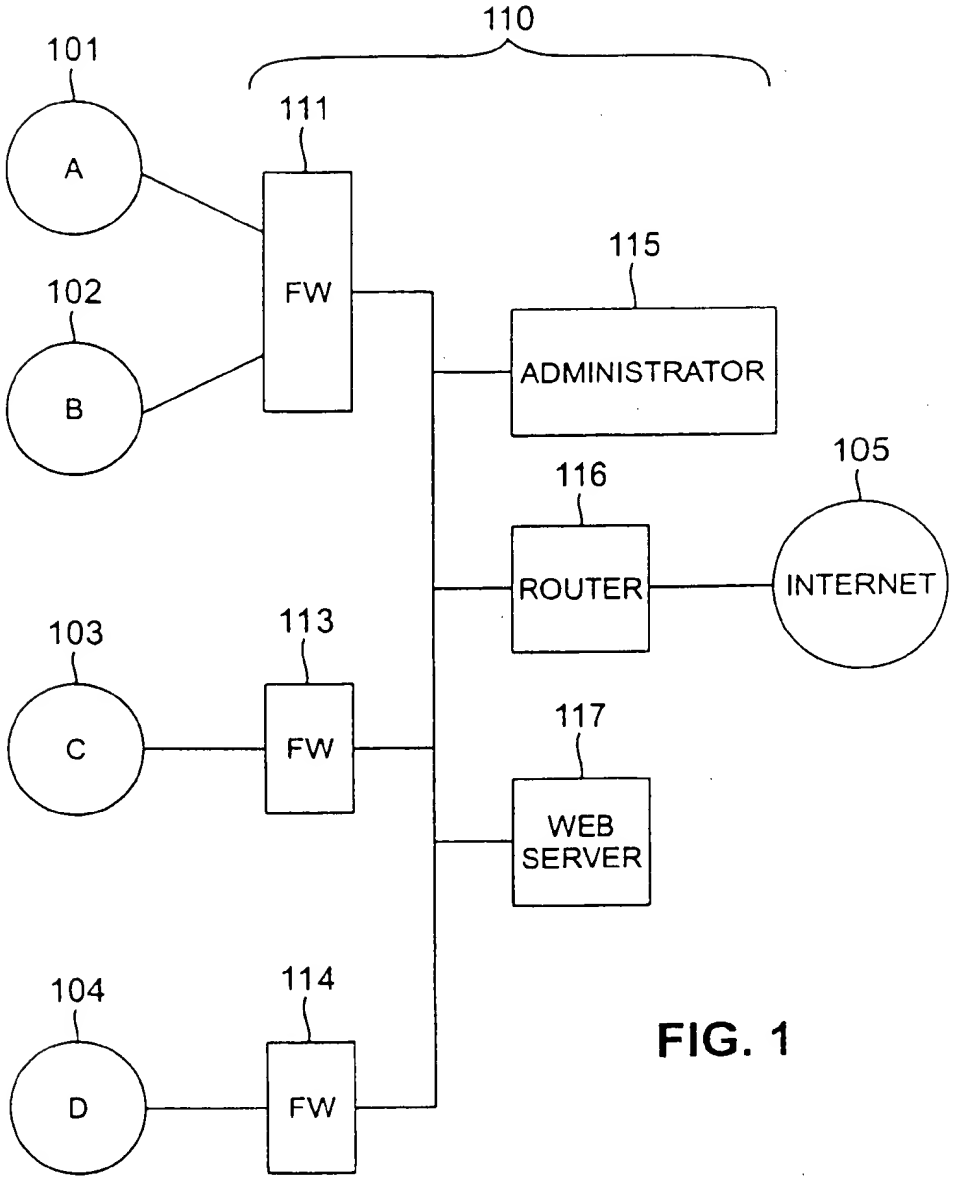


FIG. 1

| RULE NO. | SOURCE HOST | DEST. HOST | SERVICE | ACTION |
|---|---|---|---|---|
| 10 | A | B | FTP | PASS |
| 20 | A | * | * | DROP |
| 30 | B | C | TELNET | PROXY |
| 40 | * | D | MAIL | PASS |

FIG. 3

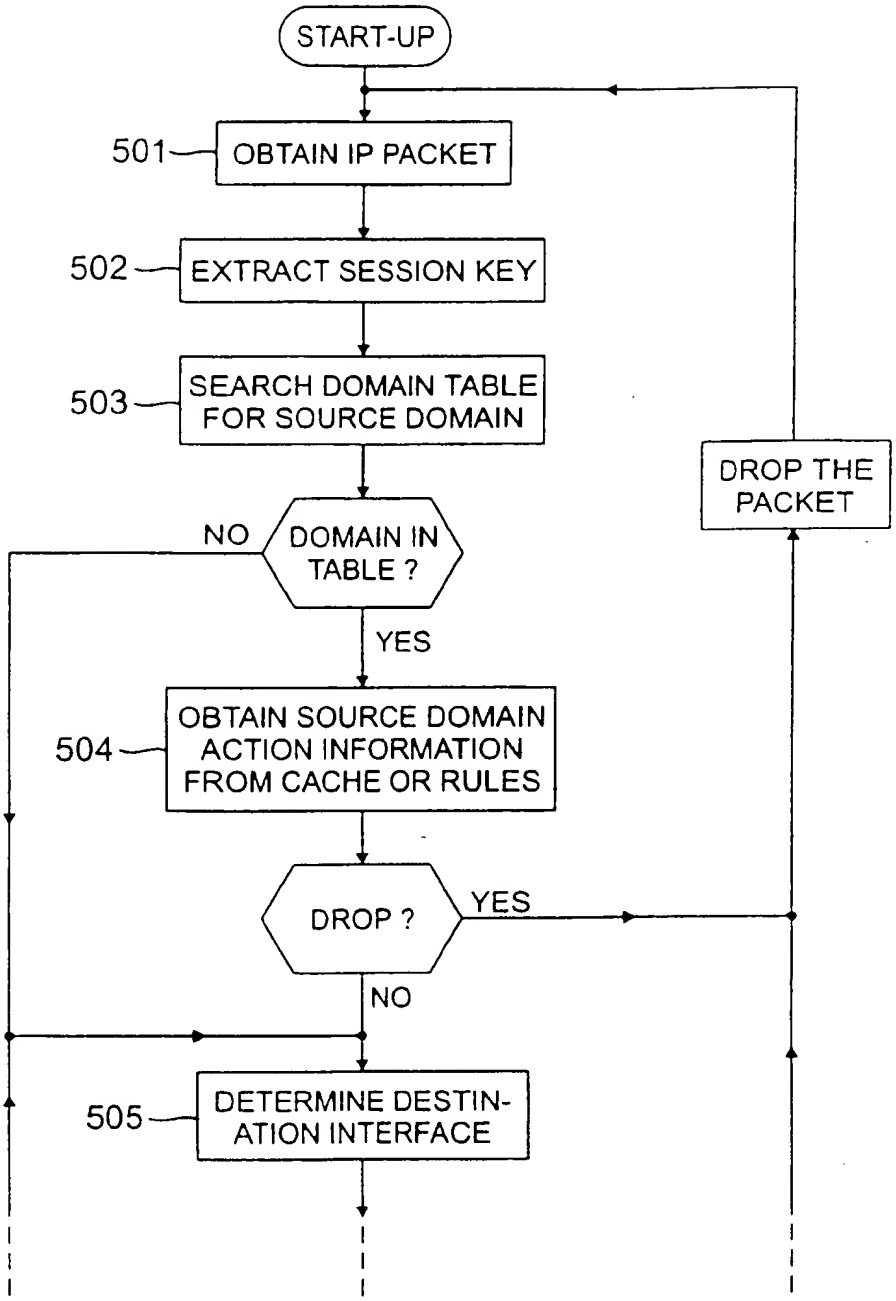


FIG. 5A

| INTERFACE | ADDRESS RANGE | DOMAIN |
|---|---|---|
| 0 | 10.50.0.0 - 10.50.255.255 | A |
| 0 | 10.60.0.0 - 10.60.255.255 | B |
| 1 | * | C |
| 2 | * | * |

FIG. 6

| NAME | SOURCE HOST | DEST. HOST | SERVICE | ACTION | HIT COUNT |
|---|---|---|---|---|---|
| TRACEROUTE | ^ | * | TRACEROUTE | PASS | 1 |
| PPTP | * | * | TCP/1723 | PASS | 1 |
| TELNET | A | B | TELNET | DROP | 10 |
| REALAUDIO | ^ | * | TCP/7070 | PASS | 1 |

FIG. 8

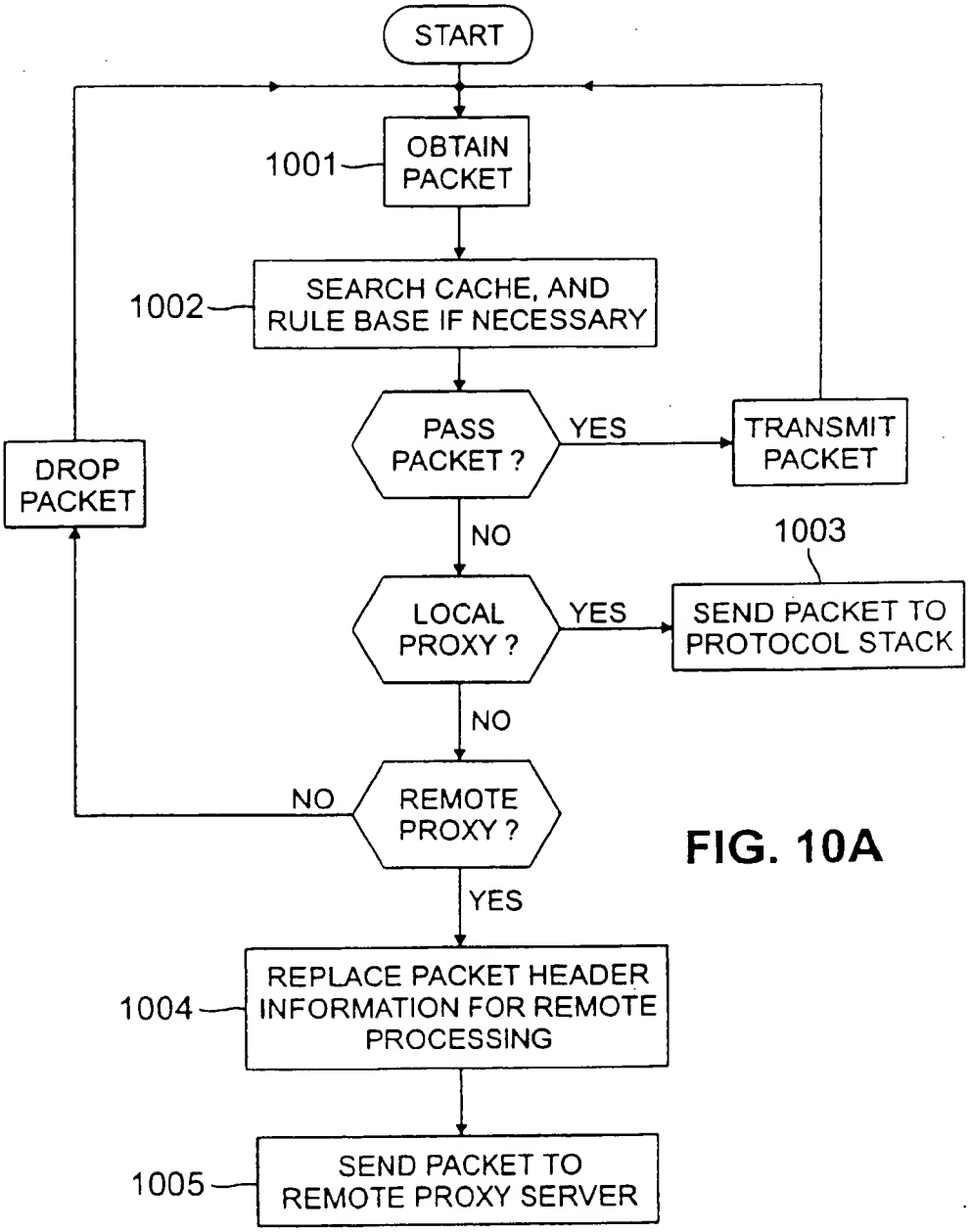


FIG. 10A

European Patent Office

# EUROPEAN SEARCH REPORT

Application Number
EP 98 30 6998

## DOCUMENTS CONSIDERED TO BE RELEVANT

| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.6) |
|---|---|---|---|
| X<br>Y | EP 0 743 777 A (SUN MICROSYSTEMS INC)<br>20 November 1996<br>* abstract *<br>* column 2, line 17-40 *<br>* column 4, line 1 - column 5, line 2 *<br>* column 6, line 20 - column 7, line 13 *<br>* column 7, line 26 - column 8, line 25 *<br>* column 10, line 44 - column 13, line 6 *<br>* figures 5,8 *<br>--- | 1-8,<br>11-19<br>9-11 | H04L29/06 |
| X<br>A<br>Y | WO 97 00471 A (DOGON GIL ;KRAMER SHLOMO (IL); SHWED GIL (IL); ZUK NIR (IL); BEN R)<br>3 January 1997<br>* abstract *<br>* page 4, line 6-24 *<br>* page 5, line 5-21 *<br>* page 15, line 13 - page 16, line 26 *<br>* page 18, line 7-14 *<br>* page 20, line 11 - page 22, line 16 *<br>* figures 4,5 *<br>--- | 1-5,8,<br>12,14,<br>16,19<br>5,7,9,<br>10,13,<br>15,17,18<br>9,11 | TECHNICAL FIELDS SEARCHED (Int.Cl.6)<br>H04L |
| E | US 5 848 233 A (PATRICK MICHAEL W ET AL)<br>8 December 1998<br>* abstract *<br>* column 2, line 56 - column 3, line 22 *<br>* column 3, line 41-50 *<br>* column 3, line 64 - column 4, line 20 *<br>* column 6, line 5 - column 7, line 16 *<br>* column 8, line 1-50 *<br>* column 9, line 33-59 *<br>* claims 1,2 *<br>---<br>-/-- | 1-8,12,<br>14-19 | |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| THE HAGUE | 29 January 1999 | Lázaro Lõpez, M.L. |

EPO FORM 1503 03.82 (P04C01)

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

# ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 98 30 6998

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on. The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

29-01-1999

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| EP 0743777 A | 20-11-1996 | US 5802320 A | 01-09-1998 |
| | | JP 9224053 A | 26-08-1997 |
| WO 9700471 A | 03-01-1997 | US 5606668 A | 25-02-1997 |
| | | AU 6135696 A | 15-01-1997 |
| | | CA 2197548 A | 03-01-1997 |
| | | EP 0807347 A | 19-11-1997 |
| | | JP 10504168 T | 14-04-1998 |
| | | NO 970611 A | 15-04-1997 |
| | | US 5835726 A | 10-11-1998 |
| | | CA 2138058 A | 16-06-1995 |
| | | EP 0658837 A | 21-06-1995 |
| | | JP 8044642 A | 16-02-1996 |
| US 5848233 A | 08-12-1998 | WO 9826555 A | 18-06-1998 |
| WO 9702734 A | 30-01-1997 | US 5751971 A | 12-05-1998 |
| | | AU 6545396 A | 10-02-1997 |

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82